IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
Before the Board of Patent Appeals and Interferences

In re Patent Application of

Atty Dkt. SCS-550-540
C# M#
Confirmation No. 4256

BUTCHER et al
TC/A.U.: 2183

Serial No. 10/807,499
Examiner: A. Li

Filed: March 24, 2004
Date: July 21, 2008

Title: COMPARE AND BRANCH MECHANISM

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

☐ **Correspondence Address Indication Form Attached.**

☐ **NOTICE OF APPEAL**
Applicant hereby **appeals** to the Board of Patent Appeals and Interferences from the last decision of the Examiner twice/finally rejecting applicant's claim(s). $510.00 (1401)/$255.00 (2401) $

☒ An appeal **BRIEF** is attached in the pending appeal of the above-identified application $510.00 (1402)/$255.00 (2402) $ 510.00

☐ Credit for fees paid in prior appeal without decision on merits -$ ( )

☐ A reply brief is attached. **(no fee)**

☒ Petition is hereby made to extend the current due date so as to cover the filing date of this paper and attachment(s)
One Month Extension $120.00 (1251)/$60.00 (2251)
Two Month Extensions $460.00 (1252)/$230.00 (2252)
Three Month Extensions $1050.00 (1253/$525.00 (2253)
Four Month Extensions $1640.00 (1254/$820.00 (2254) $ 120.00

☐ "Small entity" statement attached.

Less month extension previously paid on -$( )

TOTAL FEE ENCLOSED $ 630.00

☒ **CREDIT CARD PAYMENT FORM ATTACHED.**

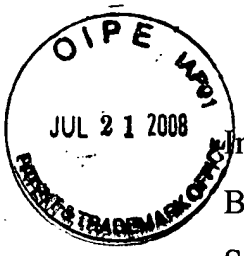
Any future submission requiring an extension of time is hereby stated to include a petition for such time extension. The Commissioner is hereby authorized to charge any deficiency, or credit any overpayment, in the fee(s) filed, or asserted to be filed, or which should have been filed herewith (or with any paper hereafter filed in this application by this firm) to our **Account No. 14-1140.** A duplicate copy of this sheet is attached.

901 North Glebe Road, 11th Floor
Arlington, Virginia 22203-1808
Telephone: (703) 816-4000
Facsimile: (703) 816-4100
SCS:kmm

NIXON & VANDERHYE P.C.
By Atty: Stanley C. Spooner, Reg. No. 27,393

Signature:

07/22/2008 SZEWDIE1 00000167 10807499
02 FC:1251 120.00 OP



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Patent Application of
BUTCHER et al
Serial No. 10/807,499
Filed: March 24, 2004
For: COMPARE AND BRANCH MECHANISM

Confirmation No.: 4256
Atty. Ref.: 550-540
Group: 2183
Examiner: A. Li

* *

APPEAL BRIEF

On Appeal From Group Art Unit 2183

Stanley C. Spooner
NIXON & VANDERHYE P.C.
11th Floor, 901 North Glebe Road
Arlington, Virginia 22203
(703) 816-4028
Attorney for Appellant

07/22/2008 SZEWDIE1 00000167 10807499
01 FC:1402 510.00 OP

TABLE OF CONTENTS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Patent Application of

BUTCHER et al — Atty. Ref.: 550-540

Serial No. 10/807,499 — Group: 2183

Filed: March 24, 2004 — Examiner: A. Li

For: COMPARE AND BRANCH MECHANISM

* *

July 21, 2008

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

I. REAL PARTY IN INTEREST

The real party in interest in the above-identified appeal is ARM Limited by virtue of an assignment of rights from the inventors to ARM Limited recorded July 27, 2004 at Reel 15620, Frame 107.

II. RELATED APPEALS AND INTERFERENCES

There are believed to be no related appeals, interferences or judicial proceedings with respect to the present application, other than the Pre-Appeal Brief Request for Review previously filed in this application on April 3, 2008.

III. STATUS OF CLAIMS

Claims 1-48 stand rejected in the Final Official Action (it will be noted that both in the Final Official Action and in the Notice of Panel Decision, the PTO erroneously indicates that claim 49 is rejected, when in fact no claim 49 has ever been offered for consideration). The Examiner contends that claims 1-48 are rejected under 35 USC §103 over various combinations of prior art references. However, as noted in the Advisory Action mailed March 28, 2008 (Paper No. 20080320), the Examiner confirms that amendments in the Rule 116 Amendment filed March 4, 2008 had been entered and "remove the objections to the claims and drawings and the rejection of claims 1-12 under 35 USC §112, second paragraph. Also, the rejections of claims 1-12 under 35 USC 101 is removed in view of the arguments."

Thus, with respect to the Final Rejection, only the rejections of claims 1-48 under 35 USC §103 remain and these rejections of claims 1-48 are appealed.

IV. STATUS OF AMENDMENTS

As noted above, an Amendment under the provisions of Rule 116 was filed March 4, 2008, and those amendments were entered by the Examiner as noted in the Advisory Action mailed March 28, 2008. Otherwise, no further response has been submitted other than the filing of a Pre-Appeal Brief Request for Review which decision was mailed on May 21, 2008.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

Appellants' specification and figures provide an explanation of the claimed invention set out in independent claims 1, 13, 25 and 37, with each claimed structure and method step addressed as to its location in the specification and in the figures.

1. Apparatus for processing data comprising:

processing logic [processor core 10 shown in Figure 1 and discussed on page 8, lines 22-30 and elsewhere in the specification] operable to perform data processing operations; and

an instruction decoder [instruction decoder 20 shown in Figure 1 and discussed on page 8, lines 22-30 and elsewhere in the specification] for decoding program instructions to control said processing logic to perform data processing operations specified by said program instructions, wherein said instruction decoder, in response to a compare and branch instruction [CHKA.X in instruction pipeline 22 in Figure 1 and discussed on page 9, lines 24-26 and elsewhere in the specification], comprises a decoder [20] for:

(i) performing a comparison between a first value stored in a first register and a second value stored in a second register [steps 50 and 52 as shown in Figure

4 and discussed on page 12, line 26 to page 13, line 1 and elsewhere in the specification];

(ii) copying, in dependence upon a result of said comparison, a program counter value to a third register [steps 58 as shown in Figure 4 and discussed on page 13, lines 3-4 and elsewhere in the specification];

(iii) determining a target branch address from a pre-programmed stored value and said program counter value [step 60 as shown in Figure 4 and discussed on page 13, lines 5-9 and elsewhere in the specification]; and

(iv) branching to a sub-routine at said target branch address in dependence upon a result of said comparison [step 62 as shown in Figure 4 and discussed on page 13, lines 9-12 and elsewhere in the specification].

13. A method of processing data with an apparatus for processing data having processing logic [processor core 10 shown in Figure 1 and discussed on page 8, lines 22-30 and elsewhere in the specification] operable to perform data processing operations and an instruction decoder [instruction decoder 20 shown in Figure 1 and discussed on page 8, lines 22-30 and elsewhere in the specification] operable to decode a compare and branch instruction to control said processing logic to perform data processing operations specified by said program instructions, said method comprising the steps of:

(i) performing a comparison between a first value stored in a first register and a second value stored in a second register [steps 50 and 52 as shown in Figure 4 and discussed on page 12, line 26 to page 13, line 1 and elsewhere in the specification];

(ii) copying a program counter value, in dependence upon said comparison, to a third register [steps 58 as shown in Figure 4 and discussed on page 13, lines 3-4 and elsewhere in the specification];

(iii) determining a target branch address from a pre-programmed stored value and said program counter value [step 60 as shown in Figure 4 and discussed on page 13, lines 5-9 and elsewhere in the specification]; and

(iv) branching to a sub-routine at said target branch address in dependence upon a result of said comparison [step 62 as shown in Figure 4 and discussed on page 13, lines 9-12 and elsewhere in the specification].

25. A computer program product comprising a computer-readable storage medium including a computer program operable to control an apparatus for processing data having processing logic [processor core 10 shown in Figure 1 and discussed on page 8, lines 22-30 and elsewhere in the specification] operable to perform data processing operations and an instruction decoder [instruction decoder 20 shown in Figure 1 and discussed on page 8, lines 22-30 and elsewhere in the

specification] operable to decode a compare and branch instruction to control said processing logic to perform data processing operations specified by said program instructions, said computer program comprising the steps of:

(i) performing a comparison between a first value stored in a first register and a second value stored in a second register [steps 50 and 52 as shown in Figure 4 and discussed on page 12, line 26 to page 13, line 1 and elsewhere in the specification];

(ii) copying a program counter value, in dependence upon said comparison, to a third register [steps 58 as shown in Figure 4 and discussed on page 13, lines 3-4 and elsewhere in the specification];

(iii) determining a target branch address from a pre-programmed stored value [step 60 as shown in Figure 4 and discussed on page 13, lines 5-9 and elsewhere in the specification]; and

(iv) branching to a sub-routine at said target branch address in dependence upon a result of said comparison [step 62 as shown in Figure 4 and discussed on page 13, lines 9-12 and elsewhere in the specification].

37. A computer program product comprising a computer-readable storage medium including a computer program [shown in Figure 2 and discussed on page 10, line 9 to page 11, line 5 and elsewhere in the specification] operable to

translate non-native program instructions to form native program instructions directly decodable by an apparatus for processing data having processing logic [processor core 10 shown in Figure 1 and discussed on page 8, lines 22-30 and elsewhere in the specification] operable to perform data processing operations and an instruction decoder [instruction decoder 20 shown in Figure 1 and discussed on page 8, lines 22-30 and elsewhere in the specification] operable to decode a compare and branch instruction to control said processing logic to perform data processing operations specified by said program instructions, said native program instructions comprising:

(i) performing a comparison between a first value stored in a first register and a second value stored in a second register [steps 50 and 52 as shown in Figure 4 and discussed on page 12, line 26 to page 13, line 1 and elsewhere in the specification];

(ii) copying a program counter value, in dependence upon said comparison, to a third register [steps 58 as shown in Figure 4 and discussed on page 13, lines 3-4 and elsewhere in the specification];

(iii) determining a target branch address from a pre-programmed stored value [step 60 as shown in Figure 4 and discussed on page 13, lines 5-9 and elsewhere in the specification]; and

(iv) branching to a sub-routine at said target branch address in dependence upon a result of said comparison [step 62 as shown in Figure 4 and discussed on page 13, lines 9-12 and elsewhere in the specification].

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-5, 7, 9-11, 13-17, 19, 21-23, 25-29, 31, 33-35, 37-41, 43 and 45-47 stand rejected under 35 USC §103 as unpatentable over Ishizaki (U.S. Patent 6,484,314) in view of Hennessy ("Computer Organization and Design: The Hardware/Software Interface").

Claims 6, 18, 30 and 42 stand rejected under 35 USC §103 as being unpatentable over Ishizaki in view of Hennessy in view of Assembly Programming ("The Art of Assembly Programming").

Claims 8, 20, 32 and 44 stand rejected under 35 USC §103 as unpatentable over Ishizaki in view of Hennessy in view of Schmidt (U.S. Patent 5,727,227).

Claims 12, 24, 36 and 48 stand rejected under 35 USC §103 as unpatentable over Ishizaki in view of Hennessy in further view of Wikipedia (the Wikipedia term relating to "protected mode").

VII. ARGUMENT

Appellants' arguments include the fact that the burden is on the Examiner to first and foremost properly construe the language of the claims to determine

what structure and/or method steps are covered by that claim. After proper construction of the claim language, the burden is also on the Examiner to demonstrate where a single reference (in the case of anticipation) or a plurality of references (in the case of an obviousness rejection) teaches each of the structures and/or method steps recited in independent claims 1, 13, 25 and 37.

Furthermore, the Court of Appeals for the Federal Circuit has stated in the case of *In re Rouffet*, 47 USPQ2d 1453, 1458 (Fed. Cir. 1998)

> to prevent the use of hindsight based on the invention to defeat patentability of the invention, this court **requires** the examiner to show a **motivation** to combine the references that create the case of obviousness. In other words, the Examiner **must show reasons** that the skilled artisan, confronted with the same problems as the inventor and with no knowledge of the claimed invention, would select the elements from the cited prior art references for combination in the manner claimed. (Emphasis added).

In its recent decision, the U.S. Supreme Court in *KSR International Co. v. Teleflex Inc.*, 82 USPQ2d 1385 (April 2007), held that it is often necessary for a court to look to interrelated teachings of multiple patents, the effects of demands known to the design community or present in the marketplace and the background knowledge possessed by a person of ordinary skill in the art in order to determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue. The Supreme Court held that "[t]o facilitate review, this analysis should be made explicit." *Id.* at 1396.

The Supreme Court in its *KSR* decision went on to say that it followed the Court of Appeals for the Federal Circuit's advice that "rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness" (the Supreme Court quoting from the Court of Appeals for the Federal Circuit in *In re Kahn*, 78 USPQ2d 1329 (Fed. Cir. 2006)).

A. The Examiner's Various Admissions Are Appreciated

The Examiner admits that Ishizaki does not teach the "copying, in dependence upon . . . ;" or the "determining a target branch address . . . ;" which are steps for the decoder in claim 1 (Final, page 6, section 13, first sentence). Also, she admits that Ishizaki fails to disclose the claimed specifics "on how exception handling affects the program counter" (page 6, section 13, second sentence) which is taken as an admission of Ishizaki's failure to teach the claimed "branching . . . in dependence upon said comparison." These admissions are appreciated.

B. The Examiner's allegation that Hennessey teaches the claimed "copying" and "determining" steps is unsupported and incorrect

Hennessy at page 411, lines 22-25, specifically states that "[t]he basic action that the machine must perform when an exception occurs is to save the address of the offending instruction in the exception program counter (EPC) and

then transfer control to the operating system." In Hennessy, the program counter value (address of the offending instruction) is copied to an exception program counter not in response to a comparison result (as required by the claimed "comparison" in Appellants' claim 1), but rather, in response to an exception. Hence, Hennessey cannot teach the claimed "copying, in dependence upon a result of said comparison, . . ." or the "branching . . . in dependence upon a result of said comparison."

Quite clearly, Hennessy does not teach the subject matter of claims 1, 13, 25 and 37 sections (ii) or (iii).

C. The Examiner fails to appreciate that the combination of Ishizake and Hennessy would not disclose all features of the independent claims

As noted above in Section A, the Examiner admits that the claimed features of "copying," "determining" and "branching" are missing from Ishizaki. Also as noted above in Section B, Hennessy cannot teach the "copying" or "branching" features because it does not perform the "comparison." As a result, even if the Ishizake and Hennessy references were combined, they fail to teach the features in the independent claims.

Accordingly, there can be no *prima facie* case of obviousness under 35 USC §103 with respect to independent claims 1, 13, 25 & 37 or claims dependent thereon.

D. The Examiner fails to provide any reason or motivation for combining references

In its recent decision, the U.S. Supreme Court in *KSR International Co. v. Teleflex Inc.*, 82 USPQ2d 1385 (April 2007), held that "rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness" (emphasis added).

The only rationale is on page 7 of the Final where the Examiner merely concludes that "it would be obvious . . . to incorporate copying the program value and determining the branch target address from the program counter value to ensure the exception handler takes the appropriate action to report and correct the error and restart program execution when the exception is handled." This is precisely the sort of "conclusory" statement that the Supreme court has held is insufficient to establish a case of obviousness.

E. The Examiner fails to recognize that she has misunderstood the Hennessy reference

The Examiner contends that it would be obvious to one of ordinary skill in the art in view of Hennessy "to ensure the exception handler takes the appropriate action to report and correct the error and restart program execution when the exception is handled." (page 7, last sentence in section 13). While not suggested in Hennessy, this functionality is disclosed in Ishizaki which is directed to

achieving this result (but in a very different manner from Appellants' independent claims). Ishizaki teaches at column 5, lines 51-57, that "when a condition described in the tw/twi [compare and branch] instructions is established, and the processing branches to the exception handler, instructions . . . are examined to detect the type of exception that has occurred."

Thus, the Examiner has confused that which is taught by Ishizaki (in a completely different context) with that which is not suggested by Hennessy.

F. The Examiner fails to recognize that Ishizaki teaches away from the claimed invention

In view of the above, Ishizaki (and not Hennessy) clearly teaches that the instruction (that gave rise to the exception) is decoded to determine the cause of the exception. In this way, Ishizaki is able to "report and correct the error." This teaching is central to the operation of the Ishizaki device and one of ordinary skill in the art would not contemplate replacing these critical features without some good reason or strong motivation. These features of Ishizaki clearly would lead one of ordinary skill in the art away from "copying, in dependence upon a result of said comparison, a program counter value to a third register" as in claim 1, section (ii) and "determining a target branch address from a pre-program stored value and said program counter value" as recited in sub-section (iii) of claim 1.

Thus, Ishizaki would clearly lead one of ordinary skill in the art away from the claims "copying" and "branching" steps that both depend upon the "result of

said comparison" and thus any *prima facie* case of obviousness had been rebutted by the adverse teaching.

G. The Examiner fails to recognize that Ishizaki and Hennessy are mutually incompatible

It is possible that the Patent Office confusion is because the Examiner has not appreciated that Ishizaki and Hennessy relate to the handling of two very different types of exceptions. Ishizaki, like the present invention, relates to the handling of "**software exceptions**," especially where the software program itself handles any events which disrupt the normal flow of the execution of the program. Hennessy relates to the handling of **CPU exceptions** whereby an unexpected event within the processor is analyzed and dealt with by the operating system.

The inherent incompatibility between the Ishizaki and Hennessy references is clearly highlighted by the definition of "exceptions and interrupts" given on page 410, lines 24 and 25 of Hennessy which he defines as "events other than branches or jumps that change the flow of instruction execution." This is in clear conflict with the Ishizaki reference which specifically deals with an "exception instruction" which compares values and then branches to an exception handler in dependence on the result of the comparison.

The Examiner's only response, in the Advisory Action, is to allege the Hennessy quote is "taken out of context" but she does not evidence and support for this position. Importantly, the Examiner does admit that Hennessy's exception

is "an unexpected event **from within the processor**" (the CPU) and this is completely compatible with Appellant's position that Hennessy relates to **CPU exceptions** and not Ishizaki's **software exceptions**.

The Examiner, in the Advisory Action and in spite of her admission above, continues to assert that Hennessy relates to software exceptions. However, she mistakenly asserts that MIPS is a software language (when in fact it is a RISC architecture – see the Hennessy comparison of architectures on page 410, line 32) and doesn't suggest a software language. The mutual incompatibility of the two references teachings rebuts any *prima facie* case of obviousness.

H. The Examiner has failed to establish a *prima facie* case of obviousness of the independent claims over the Ishizaki/Hennessy combination by itself or in further view of any other cited prior art

The Examiner has made several admissions with respect to failures of the Ishizaki reference to teach specifically recited structures and/or method steps set out in the independent claims as noted above in Section A. The Examiner's allegation that those structures and method steps which are admittedly missing from Ishizaki are actually shown in Hennessy is simply unsupported by the Hennessy reference, and thus is incorrect. This is discussed in detail in Section B above. As a result of Sections A and B, it is clear that even if combined, the Ishizaki and Hennessy references do not disclose all features of the independent claims as noted in Section C above. As a result, the Examiner fails to meet her burden of establishing a

prima facie case of obviousness by showing that each claimed element and/or method step and each claimed interrelationship between elements or method steps is disclosed in the Ishizaki/Hennessy combination of references. The failure to meet the burden of establishing a *prima facie* case of obviousness with respect to the Ishizaki/Hennessy combination means that there is no basis for rejecting any independent claim or any claims dependent thereon over the Ishizaki/Hennessy combination.

Additionally, the Examiner fails to allege that any secondary reference, such as Assembly Programming, Schmidt or Wikipedia, contains any teaching of the copying and determining elements and method steps admitted by the Examiner to be missing from Ishizaki and unsupported by any correct reference to the Hennessy reference. If the claimed elements are not shown in the primary combination of Ishizaki/Hennessy and are not even alleged to be disclosed in the secondary references, the Examiner has clearly failed to set out a *prima facie* case of obviousness of all claims, regardless of which secondary references are combined with the Ishizaki/Hennessy combination.

In view of the above, the Examiner has simply failed to meet her burden of establishing a *prima facie* case of obviousness.

Additionally, as discussed in Section D above, the Examiner has failed to provide any analysis of how or why one of ordinary skill in the art would combine the references. As the U.S. Supreme Court has held in the *KSR* case noted above,

"to facilitate this review [of the examiner's apparent reason for combining references], this analysis should be made explicit." Rather than providing the required explicit "analysis," the Examiner merely makes conclusory statements suggesting that one of ordinary skill in the art would want to modify the Ishizaki reference in the manner of Hennessy. The failure to provide the required "explicit" analysis is a further indication that the Examiner has simply failed to meet her burden of establishing a *prima facie* basis for the case of obviousness and therefore any further rejection thereunder is respectfully traversed.

In fact, as noted in Section E above, the Examiner actually appears to misunderstand the teaching in the Hennessy reference. As noted in Section E, the Examiner has confused material taught by the Ishizaki reference (in a completely different context) with material which is not suggested or taught in the Hennessy reference.

Moreover, the Examiner apparently fails to recognize that the Ishizaki reference actually teaches away from the claimed invention, as discussed in Section F. As noted above, the cited features of Ishizaki would lead one of ordinary skill in the art away from Appellants' claimed copying and determining steps and structures. As a result, because the cited references would lead one of ordinary skill in the art away from the claimed combination, Appellants have rebutted any *prima facie* case of obviousness made out by the Examiner (although

Appellants strongly argue that no *prima facie* case has as yet been made out by the Examiner).

Finally, as noted in Section G above, the Examiner fails to appreciate that the two primary references, i.e., Ishizaki and Hennessy, are mutually incompatible. Hennessy relates to the handling of CPU exceptions and Ishizaki relates to software exceptions. The Examiner does admit in the Advisory Action comments that Hennessy's exception is a CPU exception and that Ishizaki's exceptions are software exceptions. The Examiner apparently fails to appreciate that these are mutually incompatible and therefore this is further evidence that the two references cannot be combined, thereby rebutting any *prima facie* case of obviousness which has been made (and, as noted above, Appellants' view is that the Examiner has simply failed to establish any *prima facie* case of obviousness).

Accordingly, each of the independent claims 1, 13, 25 and 37 are clearly patentable over the Ishizaki/Hennessy combination by itself or when combined with any one of the other cited references and any further rejection thereunder is respectfully traversed.

VIII. CONCLUSION

In the Final Rejection, the Examiner admits that at least three claimed aspects of the current invention are not disclosed in the Ishizaki reference, and fails to disclose how or why she believes them to be disclosed in the Hennessy

reference. Accordingly, even if the references were combined, there can be no *prima facie* case of obviousness. Additionally, the Examiner fails to provide any "explicit" rationale for picking and choosing elements from the prior art and then combining them in the manner of the independent claims. She further appears to disregard both Ishizaki's "teaching away" and the mutual incompatibility between Ishizaki and Hennessy, both of which rebut any case of obviousness under 35 USC §103. Hennessy, which deals with **CPU exceptions** as events other than branches or jumps is not combinable with Ishizaki which deals with **branch instructions**.

As a result of the above, there is simply no support for the rejection of Applicants' independent claims or claims dependent thereon under 35 USC §103. Thus, and in view of the above, the rejection of claims 1-48 under 35 USC §103 is clearly in error and reversal thereof by this Honorable Board is respectfully requested.

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: _____

Stanley C. Spooner
Reg. No. 27,393

SCS:kmm
Enclosure

IX. CLAIMS APPENDIX

1. Apparatus for processing data comprising:

processing logic operable to perform data processing operations; and

an instruction decoder for decoding program instructions to control said processing logic to perform data processing operations specified by said program instructions, wherein said instruction decoder, in response to a compare and branch instruction, comprises a decoder for:

(i) performing a comparison between a first value stored in a first register and a second value stored in a second register;

(ii) copying, in dependence upon a result of said comparison, a program counter value to a third register;

(iii) determining a target branch address from a pre-programmed stored value and said program counter value; and

(iv) branching to a sub-routine at said target branch address in dependence upon a result of said comparison.

2. The apparatus as claimed in claim 1, wherein said instruction is an array bounds checking instruction and said sub-routine is an array bounds exception handling routine.

3. The apparatus as claimed in claim 1, wherein at least one of said first register and said second register are specified within said compare and branch instruction.

4. The apparatus as claimed in claim 2, wherein said first value is a reference value specifying an array size and said second value is a test value determined from a decoded program instruction.

5. The apparatus as claimed in claim 4, wherein said comparison determines whether said reference value is greater than or equal to said test value.

6. The apparatus as claimed in claim 4, wherein said result of said comparison is determined from a carry flag value and zero flag value.

7. The apparatus as claimed in claim 2, wherein said branching operation comprises copying a pointer to said array bounds exception handling routine into a register specifying a next program instruction.

8. The apparatus as claimed in claim 1, wherein said data processing apparatus comprises a co-processor and said pre-programmed stored value is read from a register of said co-processor.

9. The apparatus as claimed in claim 1, wherein said compare and branch instruction is executed within a single processing cycle of said data processing apparatus when the branch is not taken.

10. The apparatus as claimed in claim 1, wherein said instruction decoder is operable to decode translated platform-independent program instructions.

11. The apparatus as claimed in claim 10, wherein said platform independent program instructions are one of :

Java bytecodes;

.net bytecodes;

MSIL bytecodes; and

CIL bytecodes.

12. The apparatus as claimed in claim 1, wherein said data processing apparatus is operable in a user mode and a privileged mode and said data processing apparatus remains in said user mode during execution of said compare and branch instruction.

13. A method of processing data with an apparatus for processing data having processing logic operable to perform data processing operations and an instruction decoder operable to decode a compare and branch instruction to control said processing logic to perform data processing operations specified by said program instructions, said method comprising the steps of:

(i) performing a comparison between a first value stored in a first register and a second value stored in a second register;

(ii) copying a program counter value, in dependence upon said comparison, to a third register;

(iii) determining a target branch address from a pre-programmed stored value and said program counter value; and

(iv) branching to a sub-routine at said target branch address in dependence upon a result of said comparison.

14. The method as claimed in claim 13, wherein said instruction is an array bounds checking instruction and said sub-routine is an array bounds exception handling routine.

15. The method as claimed in claim 13, wherein at least one of said first register and said second register are specified within said compare and branch instruction.

16. The method as claimed in claim 14, wherein said first value is a reference value specifying an array size and said second value is a test value determined from a decoded program instruction.

17. The method as claimed in claim 16, wherein said comparison determines whether said reference value is greater than or equal to said test value.

18. The method as claimed in claim 16, wherein said result of said comparison is determined from a carry flag value and a zero flag value.

19. The method as claimed in claim 14, wherein said branching operation comprises copying a pointer to said array bounds exception handling routine into a register specifying a next program instruction.

20. The method as claimed in claim 13, wherein said data processing apparatus comprises a co-processor and said pre-programmed stored value is read from a register of said co-processor.

21. The method as claimed in claim 13, wherein said compare and branch instruction is executed within a single processing cycle of said data processing apparatus when the branch is not taken.

22. The method as claimed in claim 13, wherein said instruction decoder is operable to decode translated platform-independent program instructions.

23. The method as claimed in claim 22, wherein said platform independent program instructions are one of :

Java bytecodes;

.net bytecodes;

MSIL bytecodes; and

CIL bytecodes.

24. The method as claimed in claim 13, wherein said data processing apparatus is operable in a user mode and a privileged mode and said data processing apparatus remains in said user mode during execution of said compare and branch instruction.

25. A computer program product comprising a computer-readable storage medium including a computer program operable to control an apparatus for

processing data having processing logic operable to perform data processing operations and an instruction decoder operable to decode a compare and branch instruction to control said processing logic to perform data processing operations specified by said program instructions, said computer program comprising the steps of:

(i) performing a comparison between a first value stored in a first register and a second value stored in a second register;

(ii) copying a program counter value, in dependence upon said comparison, to a third register;

(iii) determining a target branch address from a pre-programmed stored value; and

(iv) branching to a sub-routine at said target branch address in dependence upon a result of said comparison.

26. The computer program product as claimed in claim 25, wherein said instruction is an array bounds checking instruction and said sub-routine is an array bounds exception handling routine.

27. The computer program product as claimed in claim 25, wherein at least one of said first register and said second register are specified within said compare and branch instruction.

28. The computer program product as claimed in claim 26, wherein said first value is a reference value specifying an array size and said second value is a test value determined from a decoded program instruction.

29. The computer program product as claimed in claim 28, wherein said comparison determines whether said reference value is greater than or equal to said test value.

30. The computer program product as claimed in claim 28, wherein said result of said comparison is determined from a carry flag value and a zero flag value.

31. The computer program product as claimed in claim 26, wherein said branching operation comprises copying a pointer to said array bounds exception handling routine into a register specifying a next program instruction.

32. The computer program product as claimed in claim 25, wherein said data processing apparatus comprises a co-processor and said pre-programmed stored value is read from a register of said co-processor.

33. The computer program product as claimed in claim 25, wherein said compare and branch instruction is executed within a single processing cycle of said data processing apparatus when the branch is not taken.

34. The computer program product as claimed in claim 25, wherein said instruction decoder is operable to decode translated platform-independent program instructions.

35. The computer program product as claimed in claim 34, wherein said platform independent program instructions are one of :

Java bytecodes;

.net bytecodes;

MSIL bytecodes; and

CIL bytecodes.

36. The computer program product as claimed in claim 25, wherein said data processing apparatus is operable in a user mode and a privileged mode and said data processing apparatus remains in said user mode during execution of said compare and branch instruction.

37. A computer program product comprising a computer-readable storage medium including a computer program operable to translate non-native program instructions to form native program instructions directly decodable by an apparatus for processing data having processing logic operable to perform data processing operations and an instruction decoder operable to decode a compare and branch instruction to control said processing logic to perform data processing operations specified by said program instructions, said native program instructions comprising:

(i) performing a comparison between a first value stored in a first register and a second value stored in a second register;

(ii) copying a program counter value, in dependence upon said comparison, to a third register;

(iii) determining a target branch address from a pre-programmed stored value; and

(iv) branching to a sub-routine at said target branch address in dependence upon a result of said comparison.

38. The computer program product as claimed in claim 37, wherein said instruction is an array bounds checking instruction and said sub-routine is an array bounds exception handling routine.

39. The computer program product as claimed in claim 37, wherein at least one of said first register and said second register are specified within said compare and branch instruction.

40. The computer program product as claimed in claim 38, wherein said first value is a reference value specifying an array size and said second value is a test value determined from a decoded program instruction.

41. The computer program product as claimed in claim 40, wherein said comparison determines whether said reference value is greater than or equal to said test value.

42. The computer program product as claimed in claim 40, wherein said result of said comparison is determined from a carry flag value and a zero flag value.

43. The computer program product as claimed in claim 38, wherein said branching operation comprises copying a pointer to said array bounds exception handling routine into a register specifying a next program instruction.

44. The computer program product as claimed in claim 37, wherein said data processing apparatus comprises a co-processor and said pre-programmed stored value is read from a register of said co-processor.

45. The computer program product as claimed in claim 37, wherein said compare and branch instruction is executed within a single processing cycle of said data processing apparatus when the branch is not taken.

46. The computer program product as claimed in claim 37, wherein said instruction decoder is operable to decode translated platform-independent program instructions.

47. The computer program product as claimed in claim 46, wherein said platform independent program instructions are one of :

Java bytecodes;

.net bytecodes;

MSIL bytecodes; and

CIL bytecodes.

48. The computer program product as claimed in claim 37, wherein said data processing apparatus is operable in a user mode and a privileged mode and

said data processing apparatus remains in said user mode during execution of said compare and branch instruction.

X. EVIDENCE APPENDIX

None.

XI. RELATED PROCEEDINGS APPENDIX

None.